09/402144

420 Rec'd PCT/PTO 29 SEP 1999

Ins a'

~~Description~~ 1/PRTS

~~Method and arrangement for forming and checking a checksum for digital data which are grouped into a number of data segments~~

SUB B1

~~In digital communications, i.e., during the exchange of digital data, it is frequently desirable to protect various aspects ~~protect the~~ transmission of the electronic data ~~with respect to the most varied aspects~~.~~

~~One ~~A~~ very significant aspect is the protection of integrity of the digital data to be transmitted against unauthorized modification~~, the so-called protection of the integrity of the data~~.~~

Description of the Related Art

As a protection against unauthorized modification of digital data, the a ~~so-called~~ cryptographic checksum, such as ~~for example~~ the digital signature, is known from a2 ~~[1]~~. The method described in Stallings ~~[1]~~ is based on forming a hashing value from the digital user data and the subsequent cryptographic processing of the hashing value by way ~~means~~ of a cryptographic key. The result is a cryptographic checksum. To check the integrity of the data, a corresponding cryptographic key is used for performing the inverse cryptographic operation on the checksum formed and the result is compared with the hashing value again calculated from the user data. The integrity of the user data is ensured when the hashing values match ~~are matched~~.

Ins a2

This known ~~previously~~ customary procedure requires ~~necessitates that~~ the complete user data to ~~must~~ be present on the receiver side in the identical order in which they were present when the hashing value was formed; if it is not ~~since otherwise~~ the formation of the hashing value leads to an erroneous ~~errored~~ value. In digital communications, however, it is frequently customary to subdivide and to transmit the user data to be transmitted in relatively small data segments, which are

also called data packets, due to protocol boundary conditions.

The data segments are frequently not tied to a defined order; it may not be ~~or it is not~~ possible to guarantee a defined sequential arrival of the data segments. The ~~In the~~ method described in Stallings requires ~~from [1], it is therefore required for~~ the complete user data to be reassembled again on the receiver side, ~~that is to say~~ after the transmission of the data segments, in the order in which they were originally sent. The data to be transmitted can only be verified in this order. However, this frequently means considerable additional overhead ~~expenditure~~ for the flow control of the data segments inasmuch as this is even possible at all within the framework of the protocol used.

Ins. a3

Commutative ~~From [2], commutative~~ operations are known. ~~In~~ Kiyek & Schwarz include ~~[2],~~ a general definition for commutative operations, in which ~~is also specified. Illustratively, a commutative operation~~ can be understood to be an operation in which the order of individual operations is unimportant and any order ~~each order~~ of individual operations ~~operation~~ always leads to the same total operation. A commutative operation can be, for example, an exclusive OR (EXOR) ~~EXOR~~ operation, an additive operation or also a multiplicative operation.

A ~~From [3], a~~ method and a device for generating check code segments for the occurrence of source data and for determining errors in the source data are known.

Ins a4

SUMMARY OF THE INVENTION

The invention is thus based on the object of specifying methods and arrangements for forming and checking a first commutative checksum for digital data which are grouped into a number of data segments, in which a flow control for the individual data segments is no longer required.

Ins a5

~~The object is achieved by the method according to Claim 1, by the method according to Claim 2, by the method according to Claim 3, by the arrangement according to Claim~~

~~11, by the arrangement according to Claim 12 and by the arrangement according to Claim 13.~~

In the first method ~~according to Claim 1~~, a first segment checksum is formed for each data segment for digital data which are grouped into a number of data segments. The first segment checksums formed are combined by a commutative operation to form a first commutative checksum.

In the second method ~~according to Claim 2~~, a predetermined first commutative checksum, which is allocated to digital data which are grouped into a number of data segments, is checked. This is done by a second segment checksum being formed for each data segment and a second commutative checksum being formed by a commutative operation on the second segment checksum. The second commutative checksum and the first commutative checksum are checked for a match.

In the third method ~~according to Claim 3~~ for forming and checking a first commutative checksum for digital data which is grouped into data segments, a first segment checksum is formed for each data segment and the first data checksums are combined by a commutative operation to form a first commutative checksum. For each data segment of the digital data to which the first commutative checksum is allocated, second segment checksums are formed and a second commutative checksum is formed by commutative operation on the second segment checksums. The second commutative checksum and the first commutative checksum are checked for a match.

The arrangement according to Claim 11 exhibits an arithmetic and logic unit which is arranged in such a manner that a segment checksum is formed for each data segment and that the first commutative checksum is formed by a commutative operation on the segment checksums.

a

The second arrangement ~~according to Claim 12~~ has ~~exhibits~~ an arithmetic and logic unit which is arranged in such a manner that a second segment checksum is formed for each data segment, a second commutative checksum is formed by a commutative operation on the second segment checksums, and the second commutative checksum (KP2) is checked for a match with the first commutative checksum (KP1).

a

The third arrangement ~~according to Claim 13~~ has ~~exhibits~~ an arithmetic and logic unit which is arranged in such a manner that the following method steps are performed:

a) a segment checksum is formed for each data segment,

b) the first commutative checksum is formed by a commutative operation on the segment checksums,

c) a second segment checksum is formed for each data segment of the digital data to which the first commutative checksum is allocated,

d) a second commutative checksum is formed by a commutative operation on the second segment checksums, and

e) the second commutative checksum is checked for a match with the first commutative checksum.

A considerable advantage of the methods and of the arrangements can be seen in the fact that, by using a commutative operation for individual checksums of the data segments, a flow control for the order of the individual data segments is no longer required.

Furthermore, it is no longer required to reassemble the complete user data in the original order in which the first commutative checksums were formed. The order of the individual data segments is no longer of significance in the formation of the commutative checksum.

If the digital data are transmitted between two arrangements, a further advantage of ~~the~~ these methods can be seen in ~~the fact~~ that the checking of the integrity can already be begun before all data segments have been received, since it is no longer required to maintain the original order in forming the first checksum. This leads to a timesaving in the data integrity checking ~~of the integrity of the data~~.

~~Illustratively, the~~ The invention can be seen in the fact that a checksum is formed in the case of a number of data segments which, together, form the data to be protected, and the individual checksums of the data segments are commutatively combined with one another.

Advantageous further developments of the invention are discussed below ~~obtained from the dependent claims~~.

It is advantageous to protect the first commutative checksum cryptographically by using at least one cryptographic operation.

The result of this further development is that the cryptographic security of the data is considerably increased. A cryptographic operation in this sense is, for example, the encrypting of the first commutative checksum with a symmetric or also with an assymetric encryption method which forms a cryptographic checksum. On the receiver side, the inverse cryptographic method to the cryptographic method is performed in order to ensure cryptographic security.

To form a checksum within the context of the document, various possibilities are known:

- a checksum can be formed by forming hashing values for the individual data segments;

- the checksums can also be formed by ~~so-called~~ cyclic codes (Cyclic Redundancy Check, CRC);
- a cryptographic one-way function can also be used for forming the checksums for the data segments.

The methods can be advantageously used in various application scenarios.

The methods can be used both in the transmission of digital data for protection against manipulation of the data, and in the archiving of digital data in a computer in which the first commutative checksum is formed and stored together with the data to be archived. The first commutative checksum can be checked when the digital data are loaded from the archive memory in order to detect any manipulation of the archived data.

The method can be advantageously used for protecting digital data in which the data segments ~~of which~~ are not tied to an order. Examples of such data segments are packet-oriented communication protocols, for example network management protocols such as the Simple Network Management Protocol (SNMP) or the Common Management Information Protocol (CMIP).

In the text which follows, an illustrative embodiment of the invention will be explained in greater detail with reference to a Figure. The ~~Even if the~~ illustrative embodiment ~~is~~ explained with reference to the Simple Network Management Protocol, (SNMP) in the text which follows, ~~this~~ does not imply ~~represent~~ any restriction on the applicability of the method. The method can be used whenever it is of importance to ensure integrity protection for digital data which are grouped into a number of data segments.

Ins a6>

Ins a7>

BRIEF DESCRIPTION OF THE DRAWINGS

~~The Figure shows~~ two arrangements, in which data segments ~~being~~ are transmitted from the first arrangement to the second arrangement.

In the Figure, a first computer arrangement A1, in which data segments (Di, i = 1 .. n) are stored, is shown symbolically. The data segments Di together form the digital data, which are also designated as user data, for which integrity must be maintained ~~it is of importance to ensure their integrity~~.

~~Both the~~ The first computer arrangement A1 and a second computer arrangement A2, described in the following text, ~~text which follows in each case~~ each contain an arithmetic and logic unit R which is arranged in such a manner that the method steps described below ~~in the text which follows~~ are performed.

In the first arrangement A1, the data segments Di are arranged at positions Pi within the total data stream. For each data segment Di, a first segment checksum PSi is formed ~~[lacuna]~~ by using a checksum function PSF. The individual first segment checksums ~~checksum~~ PSi are combined to form a first commutative checksum KP1 by a commutative operation as defined and described in Kiyek & Schwarz ~~[2]~~. The commutative operation on the individual checksums PSi are shown symbolically by an EXOR symbol ⊕ in the Figure.

The first commutative checksum KP1 is subjected to a cryptographic operation ~~method~~, a symmetric or asymmetric method, by using a first cryptographic key S (step 101). The result of the cryptographic operation is a cryptographic checksum KP.

Both the data segments Di and the cryptographic checksum KP are transmitted by a transmission medium, preferably a line or also a logical connection which is symbolically shown by a communication link UM in the Figure,

to a second arrangement A2 where they are received.

The crossing arrows of the data segments Di in the Figure indicate that, due to the transmission of the data segments Di, these are received in positions Pj (j = a .. z) which are displaced compared with the order in the first arrangement A1.

Thus, a data segment D2 at the first position P1 is received as data segment Da in the second arrangement A2. Data segment D1 is received as data segment Dc in the second arrangement. Data segment Dn is received as received data segment Db at the second position P2 in the second arrangement A2.

In accordance with the method used, either the first cryptographic key S is used for performing the inverse cryptographic operation on the cryptographic checksum KP if a symmetric encryption method is used, or a second cryptographic key S' is used if an asymmetric cryptographic method is used.

The result of the inverse cryptographic operation (step 102) is again the first commutative checksum KP1 with correct encryption and decryption.

This checksum is stored in the second arrangement A2. For the comparison of the data segments Dj, which are now received in permutated order compared with the original order during the formation of the first commutative checksum KP1, second segment checksums Psj are formed for the received data segments Dj, again using the same checksum ~~methods~~ functions PSF.

a

The resultant second [illegible] checksums PSj are again commutatively combined with one another to form a second commutative checksum KP2.

In a further comparative step 103, a check is made whether the first commutative checksum KP1 matches the second commutative checksum KP2.

If this is so, the integrity of the data segments Di, and thus the integrity of all the digital data, is ensured (step 104) if the cryptographic methods used or, respectively, the methods used for forming checksums ensure the corresponding cryptographic security.

If the first cryptographic checksum KP1 does not match the second cryptographic checksum KP2, the integrity of the data segments Di would be violated, possibly indicating ~~and~~ a manipulation of the data. ~~is found and~~ Such a condition would preferably reported to a user of the system.

The protocol data units (PDU) in SNMP are structured in such a manner that the user information (~~so-called~~ variable bindings) can contain a list of objects (object indicators, OID/value pairs). The order of the objects within a PDU is not specified so that it is possible for a permutation of the objects to occur during the transmission of the PDUs between the first computer arrangement A1 and the second computer arrangement A2. The invention now makes it possible to form a single cryptographic checksum over all objects of an SNMP PDU without having to take into consideration the order of the objects or of the PDUs.

The text below explains ~~In the text which follows~~, alternatives to the illustrative embodiment described above ~~will be explained~~.

The method for forming the checksum PSF can be, for example, a method for forming hashing values. However, methods for forming cyclic codes (Cyclic Redundancy Check, CRC) using feedback-type shift registers can also be used. In addition, cryptographic one-way functions can be used for forming the checksums PSi and, respectively, Psj.

Furthermore, the commutative operation can have the additional property of associativity.

Sub B4

Both the method for forming the checksum and the method for checking a checksum can be performed either independently of one another or ~~. However, the method for forming the checksum and the method for checking the checksum can also be~~ performed jointly.

Furthermore, the method also allows one ~~it is provided not to transmit digital data but~~ to archive the digital data, by ~~that is to say to store them~~ storing the digital data in the first arrangement A1, together with the first commutative checksum KP1. When the archived data are reused, i.e. ~~that is to say~~ when the data segments Di are loaded from the memory of the first arrangement A1, the method for checking the first commutative checksum KP1 as described above will then be performed. The first arrangement A1 and the second arrangement A2 can thus be identical.

The ~~Illustratively, the~~ invention can be seen where ~~in that in the case of~~ a number of data segments which~~,~~ together~~,~~ represent the data to be protected, a checksum is formed for each data segment and the individual checksums of the data segments are commutatively combined with one another. This makes it possible to form and to check a checksum without having to consider ~~take into consideration~~ the order of the data segments.

Ins a8

In this document, the following publications have been quoted:

[1] W. Stallings, Sicherheit in Netzwerk und Internet (Security in Network and Internet), Prentice Hall, ISBN 3-930436-29-9, pp. 203-223, 1995

[2] K.-H. Kiyek and F. Schwarz, Mathmatik für Informatiker (Mathematics for Computer Scientists), Teubner Verlag, ISBN 3-519-03277-X, pp. 11-13, 1989

[3] DE-A 2 048 365